- -> import the numpy library
- -> this is done in a Jupyter notebook

- ***-> import numpy as np***
    - -> if this doesn't work, you have to pip install numpy in the terminal
    - -> if pip doesn't work, you have to use pip3

- ***-> a = np.array([1,2,3])***
    - an array containing values
    - -> you can print this

- ***-> you can also b = np.array([9.0,8.0,7.0],[6.0,5.0,4.0])***
    - -> you can also nest arrays inside other arrays
    - -> you can make an element of an array equal to another array

- ***-> to get the dimension of the numpy array***
    - -> a.dim
    - -> this returns the number of dimensions

- ***-> a.shape <- this returns the shape of the vector***
    - -> the number of rows and columns
    - -> this is a function

- ***-> for the amount of memory the numpy array takes***
    - -> a.dtype
    - -> dtype('int32') <- the number should take up 4 bytes
    - -> you can tell it the number of bytes you want it to take
    - -> one number taks 4 bytes, even though they are small numbers
    - -> a = np.array([1,2,3], dtype='int16')
        - -> this tells the array to take up less size
        - -> it's the same array
        - -> it's just being stored with less data
        - -> you can also int32
        - -> a.itemsize <- this returns the number of bytes
    - -> you can also a.nbytes
    - -> b.itemsize
    - -> floats are going to be bigger than integers
    - -> to be efficient, you need to specify dtype so that it fits the data as tightly as possible

- ***-> question***

What will the following code print?

b = np.array([[1.0,2.0,3.0],[3.0,4.0,5.0]])
print(b)

[[1.0 2.0 3.0]
[3.0 4.0 5.0]]

[[1. 2. 3.] <- This one
[3. 4. 5.]]

[[1. 3.]
[2. 4.]

[3. 5.]